Reliable edge machine learning hardware for scientific applications

Tommaso Baldi*†
Javier Campos*
Ben Hawks*
Jennifer Ngadiuba*
Nhan Tran*
*Fermilab
†Università di Pisa

Daniel Diaz
Javier Duarte
Ryan Kastner
Andres Meza
Melissa Quinnan
Olivia Weng
UC San Diego

Caleb Geniesse
Amir Gholami
Michael W. Mahoney
UC Berkeley/LBNL/ICSI

Vladimir Loncar
Philip Harris
MIT

Joshua Agar
Shuyu Qin
Drexel University

Abstract—Extreme data rate scientific experiments create massive amounts of data that require efficient ML edge processing. This leads to unique validation challenges for VLSI implementations of ML algorithms: enabling bit-accurate functional simulations for performance validation in experimental software frameworks, verifying those ML models are robust under extreme quantization and pruning, and enabling ultra-fine-grained model inspection for efficient fault tolerance. We discuss approaches to developing and validating reliable algorithms at the scientific edge under such strict latency, resource, power, and area requirements in extreme experimental environments. We study metrics for developing robust algorithms, present preliminary results and mitigation strategies, and conclude with an outlook of these and future directions of research towards the longer-term goal of developing autonomous scientific experimentation methods for accelerated scientific discovery.

Index Terms—real-time systems, neural networks, digital simulation, reliability assesment, VLSI, VLSI testing

I. MOTIVATION

Ground-breaking science requires instruments that push sensing technology with increasing spatial and temporal resolution to explore nature at unprecedented scales and in extreme environments. This has led to a data generation explosion, with more and more data being generated in next-generation experiments. For example, particle physics experiments look for extremely rare collision events (one in a billion billion) that can answer fundamental questions about the fabric of space-time or the nature of dark matter. Alternatively, microscopy experiments take hundreds of thousands of images per second to understand material properties that can advance computing, quantum science, and basic energy research. There are many other applications in a wide range of domain sciences, including fusion, nuclear physics, neuroscience, and quantum
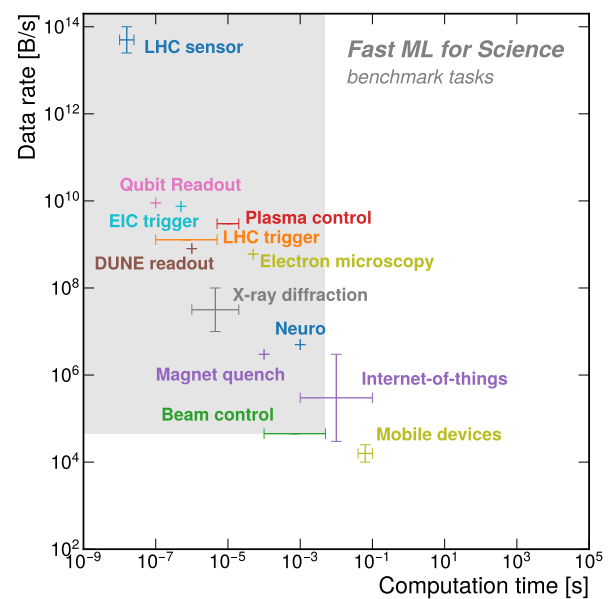
Fig. 1. Many scientific and edge ML benchmark tasks [] must process incoming data at a high rate leading to extreme low-latency and high-bandwidth requirements. Applications illustrated here range across particle physics (LHC, DUNE), nuclear physics (EIC), material science (X-ray diffraction, microscopy), neuroscience, fusion energy, quantum information science, superconducting magnet research, and particle accelerators. This can be compared against traditional internet-of-things and mobile device applications which are less stringent.

computing, that can benefit from real-time, low-latency edge processing [1].

Many of these experiments create terabytes to petabytes of data per second, and at this rate these data cannot be stored and processed with traditional methods in off-the-shelf computing clusters. Instead, scientists must process the data as close as possible to the experimental sensor—-at the edge. This is similar in some ways to autonomous vehicles and other smart sensing applications, but it occurs at unprecedented data rates and latency requirements for data processing. An illustration of this is provided in Fig. 1, which shows various benchmark tasks for different scientific applications as well as their input

bandwidth and processing latency requirements.

These challenges require efficient, specialized compute hardware at the extreme edge, e.g., FPGAs, ASICs, and systems on chip (SoCs). These platforms are common to many scientific experiments. This also has implications for the computer architectures [2] that need to meet these extreme low-latency requirements: (i) often all neural network (NN) parameters must fit on-chip; and (ii) fully on-chip inference often requires hardware-software codesign with custom/reconfigurable logic to meet latency and bandwidth constraints. Per-sensor compression and efficient aggregation of information, while preserving scientific fidelity, can significantly impact experiment data flow, analysis, control, and operation, as well as, ultimately, how quickly experiments can be performed and hypotheses explored.

In this paper, we discuss various unique and important challenges posed by deploying edge ML algorithms in realistic scientific environments with unfiltered and dynamic data streams and present proof-of-principle case studies and methodologies that address those challenges. We examine techniques:

- to validate algorithm performance in large experimental software frameworks on simulated and real data with functionally verified VLSI implementations;
- to characterize the stability of both local and global loss landscape structures in the training of highly customized and optimized algorithms; and
- to improve the tolerance of edge ML algorithms to bit flip and sensor noise faults caused by experimental conditions with ultra-fine-grained bit level inspection and targeted regularization techniques.

II. Exemplar application and previous work

For our example domain science task, we consider the CERN Large Hadron Collider (LHC) Compact Muon Solenoid (CMS) experiment [3]. This is an experiment that runs particle collision experiments that generate data rates of $\sim$40 TB/s. To reduce data rates, physicists deploy tens of thousands of ASICs and hundreds of FPGAs to make decisions on whether a specific collision is of particular interest (known as triggering) at the microsecond scale [4], [5]. One such ASIC is the endcap concentrator (ECON-T) ASICs [6], which is planned for deployment in the upcoming LHC upgrade. Each ECON-T ASIC is running a NN encoder to compress experimental data from the high-granularity endcap calorimeter (HGCAL) [7] into a smaller format for easy filtering in the trigger system. The ECON-T encoder hardware must accept new input data at 40 MHz and complete inference in 25 ns within an area budget of 4 mm^2 [6]. To meet these constraints, the ECON-T encoder model is a small two-layer NN with $\sim$2000 parameters quantized to have 6-bit fixed-point weights which operates completely on-chip. To complicate matters further, the ECON-Ts operate in a high-radiation environment (due to their close proximity to particle collisions in the LHC). High radiation causes transient hardware errors, which can lead to incorrect application output (silent data corruptions)

if the hardware is not designed robustly. The ECON-Ts filter terabytes per second of data for high-energy physics studies, and faulty execution is unacceptable. Only the NN weight parameters are vulnerable to faults because the activations are not stored in on-chip memory for longer than a cycle, as inference completes in a single cycle.

The open-source quantization-aware platform QKeras [8] is used for model training required for the algorithm development and also for generating stimuli for verification. The output of this stage is processed by the hls4ml compiler [9], [10], which translates the QKeras model description into a resultant C++ description. This is then converted to an hardware RTL description in Verilog utilizing Mentor Catapult HLS. Several verification steps are undertaken at this stage to identify bugs and improve performance: design rule checks, C simulation and code coverage, and other traditional on-chip digital verification.

The HLS model description requires approximately 1,000 lines of code. This stage is fast, $\mathcal{O}$(seconds), but it requires several hundred iterations to optimize the algorithm performance based on physics metrics. The HLS stage determines the level of parallelism in the design, choice of pipelining, resource reuse factor, and clock frequency. This directly impacts the total area, power consumption, and the latency of the design. The digital simulation and implementation stages, alternatively, is much more time intensive and can take several orders of magnitude longer [6].

III. Methodology and metrics

For ML models developed for scientific edge application, taking the ECON-T as an example, we now discuss methods developed for large scale, bit accurate simulation; metrics for robust quantized training; and approaches for mitigating effects of sensing and electronics faults.

A. Accurate and fast functional simulation

Powerful ML algorithms are valuable for real-time data processing, but they can lead to complex designs in hardware platforms such as FPGAs and ASICs. At the same time, for high data rate experiments, these algorithms need to be validated against large amounts of data offline to understand their performance in identifying rare and interesting signals amongst large background processes. This can necessitate a CPU-based inference of the model over millions or billions of events within the simulation framework of large experimental collaborations.

To that end, a firmware generation workflow like hls4ml has an extremely useful benefit that was not intended in its initial design. The hls4ml workflow emits a fuctionally *bit accurate* C representation of the NN in hardware with the aid of algorithmic C[1] and arbitrary precision[2] libraries. The generated C code contains no external dependencies and compiles with any modern C/C++ compiler, making it easy to

---

[1] https://hlslibs.org/

[2] https://docs.amd.com/r/en-US/ug1399-vitis-hls/Arbitrary-Precision-AP-Data-Types

integrate into existing simulation frameworks. This can be used to test the functional performance of the hardware algorithm in high level programming languages like Python and C much more easily and rapidly rather than in lower level hardware description languages.

For CMS trigger applications, it is necessary to emulate the ASIC and FPGA algorithms in software in order to evaluate their performance in simulation and compare data with simulation. The firmware simulation generated with hls4ml has been added to the CMS software framework (CMSSW) [11], [12] used to collect, produce, and analyze physics data. Being fully self-contained, the generated C representation is compiled as a shared library and dynamically loaded, enabling multiple model versions to simultaneously co-exist. This enables physicists to trust the performance of algorithms developed for hardware despite being tested on very different hardware platforms.

B. Quantized NN loss landscapes

Bit accurate representations of these edge ML algorithms is important because of the custom quantization often deployed to make highly optimized implementations for efficient hardware. Often training is performed for such algorithms with a Pareto optimization over the performance and the system constraints (resources, latency, area, power, etc.) [13]. However, the stability of the training is not typically considered, and the robustness of the model may depend on effects which characterize the local and global loss landscape structure. This can lead to unstable and unreliable model training and a high sensitivity to small perturbations to the input data distribution. As an illustration of this, in Figure 2, we show the loss landscapes of 4 ECON-T models trained with different uniform quantizations [14], [15].

These visualizations help us to understand the local stability of each trained model under various quantizations. To characterize these features, we explore a few metrics.

- **CKA similarity:** This metric, based on the centered kernel alignment (CKA), has shown effectiveness in capturing the correspondence between representations in various trained neural networks [16], owing to its invariance properties. Models converging to the same local minima tend to demonstrate similar characteristics, offering insights into the smoothness of the loss landscape and the potential correlation with model performance and robustness. A low value of CKA suggests a high difference between models initialized with different random parameters, implying a challenging loss landscape.
- **Hessian**: In machine learning (ML), the Hessian matrix is a square matrix that characterizes the curvature of the loss function at a specific point. The eigenvalues of the Hessian matrix provide scalar values that offer insights into the curvature type at that point. Positive eigenvalues suggest local convexity of the function, indicating a single minimum or maximum.
- **Neural Efficiency**: This metric, even if not directly related to the shape of the loss landscape, can give interesting
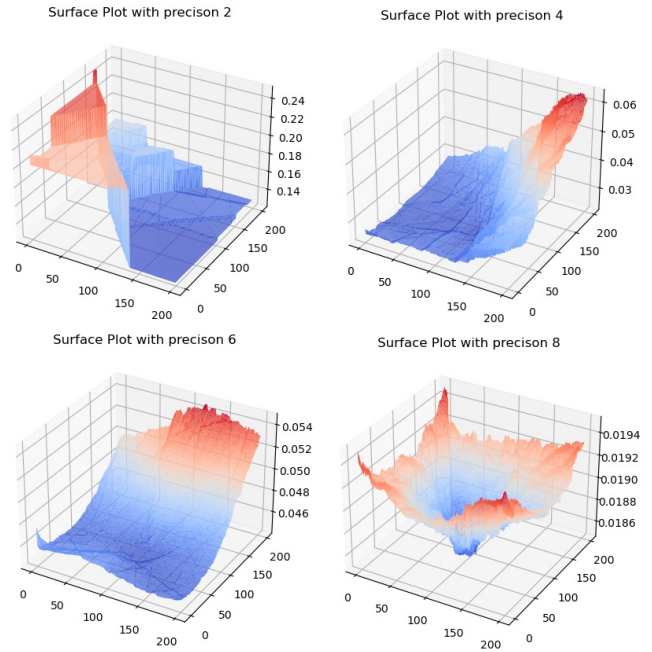

Fig. 2. ECON-T model loss landscapes illustrating varying behaviors with different uniform quantizations between 2-bit and 8-bits. A range of performance can be seen from very jagged landscapes at 2-bit weights to relatively smooth landscapes at 4- and 6-bit weights to a sharp narrow minima for 8-bit weights.

insights about the capacity of the model. Indeed, network efficiency lets us know what is the percentage of node that are effectively used during the inference of the neural network, by estimating the minimum number of neurons required to encode the information exported by the neural layer, assuming perfect encoding [17].

Figure 3 shows the outcomes of the ECON-T analysis. Performance heat maps show the performance of the autoencoder by the Earth Mover's Distance metric when 5% noise is introduced to the model inputs, Figure 3(left), revealing that the reduced precision of parameters in quantized NNs serves as regularization, guiding the model towards flat and smooth minima during training (as observed also in Figure 2). This is also reflected in the Hessian trace metric, Figure 3(right) which indicates lower bit widths have a flatter global minima.

C. Fault tolerance to bit flips and sensor noise

The final unique scientific challenge we consider is the sensitivity of the NN model to common faults: bit flip faults and sensor noise.

In Section II, we discussed that the ECON-T ASIC will operate in an extreme radiation environment, $1000\times$ that experienced in outer space. This leads to a large amount of single event upsets which cause bits to flip and which affect the performance of the ASIC. Typical radiation mitigation in hardware deploys triple modular redundancy (TMR) to triplicate the weight registers, but this incurs a 200% resource overhead. In response, we develop FKeras [18] a tool that performs ultra-fine-grained inspection of the model to study
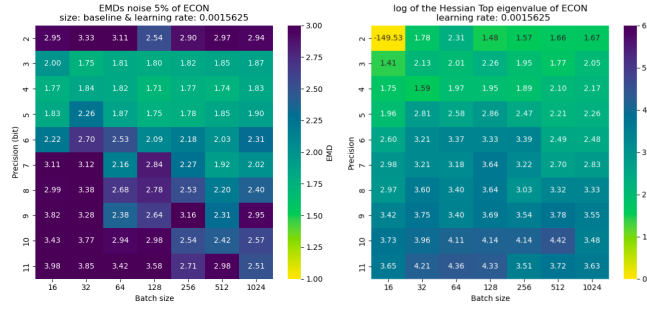
Fig. 3. Results achieved by different versions, in terms of hyperparameters, of the ECON-T model. The left heat map shows the EMD achieved by the model on noisy data. The right heat map hows the top eigenvalue, in logarithmic scale.

Fig. 4. The plot illustrates the performance of ECON-T models under 5% noisy data trained with different values of λ_{JR}, the hyperparameter used to tune the weight of the Jacobian regularization component

the bit-level sensitivity of each weight in the NN. This allows us to prioritize which bits need protection and which may be safely disregarded, reducing resource overhead.

We use FKeras [18] to perform a design space exploration on the ECON-T model with respect to model size, performance, and fault tolerance. We perform a neural architecture search of the ECON-T model, using a Bayesian optimization to find small, medium, and large quantized ECON-T models that range from less accurate to more accurate. In our FKeras analysis, we find that all of the weight bits in the small, less accurate model are sensitive to faults, i.e., flipping any bit in the weights will cause the model to perform worse. For the large, more accurate model however, we find that only 6% of the weight bits are sensitive to faults. This implies that we need only protect a small fraction of the weights [18]. The tradeoff here is that we need more resources to implement a larger model. As a result, we are met with a design tradeoff that merits careful consideration: would it be more worthwhile to implement a smaller, less accurate model that requires full protection or a larger, more accurate model that requires a small fraction of protection?

Another common fault model for sensing applications is noise which may come from the sensor itself or conditions in which it is taking data. Therefore, deploying models on the edge demands more than just achieving the desired performance metrics; it necessitates ensuring robustness to perturbations. Robust models can withstand and adapt to noisy environments, offering more reliable performance under varying input data. To meet this challenge, we employ specific mitigation techniques that leverage insights from the loss landscape analysis. The investigation into the ECON-T model highlighted the advantageous effect of converging towards wide and flat minima. To address this need, we introduced Jacobian Regularization into the training process, aiming to push decision boundaries further from individual data points by minimizing the Frobenius norm of the Jacobian matrix [19]. As showed in Figure 4, this regularization technique offers a targeted approach to enhancing model robustness, informed by the shape and characteristics of the loss landscape.
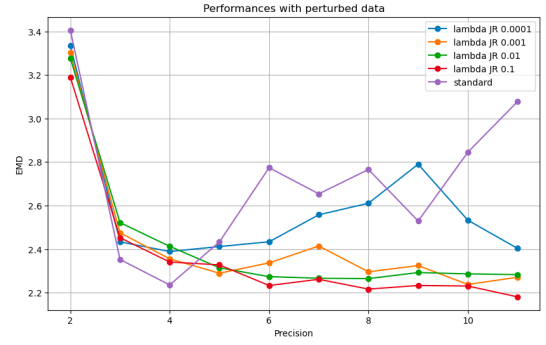
IV. Outlook

In this paper, we highlight a number of unique challenges presented by high-throughput, low-latency scientific edge ML applications to enable reliable and efficient hardware VLSI implementations. From accurate, large-scale simulation to characterizing quantization robustness to fault mitigation under bit flips and noise, we present methods that can be employed to ensure trustworthy and resilient operation of cutting-edge science experiments. The methods we discuss in this study cut across a wide range of applications beyond particle physics. Future studies will continue to study these methods on other scientific domains such as those presented in Fig. 1.

The state-of-the-art of edge ML applications in science have made significant advances in performance and hardware optimization [1]. We discuss a number of important reliability challenges, though many remain, and we call attention to the importance of designing tools, metrics, and techniques to optimize for robustness as well which will enable (semi-)autonomous, adaptive, and intelligent experimentation. Furthermore, visualization tools to aid domain scientists will enhance and transform the way experiments will operate in the future.

From the studies presented here, several interesting research lines can be extended. We would like to explore how the fault tolerance of a neural network interacts with common neural network compression techniques like pruning and quantization. Perhaps certain pruning techniques lead to more resilient neural network weights, similar to a finding in which the chosen pruning technique made a neural network more resilient to noise in its inputs [20]. It also would be worthwhile to see how loss landscapes reveal information on how resilient a neural network is to faults. While some correlation between robustness and loss landscape metrics were found, the extent of that correlation and possible causality still need to be understood. Ultimately, the design space optimization of performance, robustness, and efficiency is still in early exploration for scientific ML applications.

References

[1] A. M. Deiana, N. Tran, J. Agar, M. Blott, G. Di Guglielmo, J. Duarte, P. Harris, S. Hauck, M. Liu, M. S. Neubauer *et al.*, "Applications and techniques for fast machine learning in science," *Frontiers in big Data*, vol. 5, p. 787421, 2022.

[2] O. Weng, A. Redding, N. Tran, J. M. Duarte, and R. Kastner, "Architectural implications of neural network inference for high data-rate, low-latency scientific applications," 2024.

[3] S. Chatrchyan, G. Hmayakyan, V. Khachatryan, A. Sirunyan, W. Adam, T. Bauer, T. Bergauer, H. Bergauer, M. Dragicevic, J. Eroe *et al.*, "The cms experiment at the cern lhc," *Journal of instrumentation*, vol. 3, 2008.

[4] A. M. Sirunyan *et al.*, "Performance of the CMS Level-1 trigger in proton-proton collisions at $\sqrt{s} = 13$ TeV," *JINST*, vol. 15, no. 10, p. P10017, 2020.

[5] CMS Collaboration, "The Phase-2 Upgrade of the CMS Level-1 Trigger," CMS Technical Design Report CERN-LHCC-2020-004, CMS-TDR-021, 2020. [Online]. Available: https://cds.cern.ch/record/2714892

[6] G. Di Guglielmo, F. Fahim, C. Herwig, M. B. Valentin, J. Duarte, C. Gingu, P. Harris, J. Hirschauer, M. Kwok, V. Loncar *et al.*, "A reconfigurable neural network asic for detector front-end data compression at the hl-lhc," *IEEE Transactions on Nuclear Science*, vol. 68, no. 8, pp. 2179–2186, 2021.

[7] CMS Collaboration, "The phase-2 upgrade of the CMS endcap calorimeter," CMS Technical Design Report CERN-LHCC-2017-023, CMS-TDR-019, 2017.

[8] C. N. Coelho, A. Kuusela, S. Li, H. Zhuang, T. Aarrestad, V. Loncar, J. Ngadiuba, M. Pierini, A. A. Pol, and S. Summers, "Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors," *Nature Mach. Intell.*, vol. 3, pp. 675–686, 2021.

[9] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, J. Ngadiuba, M. Pierini, R. Rivera, N. Tran *et al.*, "Fast inference of deep neural networks in fpgas for particle physics," *Journal of Instrumentation*, vol. 13, no. 07, p. P07027, 2018.

[10] FastML Team, "fastmachinelearning/hls4ml," 2023. [Online]. Available: https://github.com/fastmachinelearning/hls4ml

[11] G. L. Bayatian *et al.*, "CMS physics: Technical design report volume 1: Detector performance and software," CMS Technical Design Report CERN-LHCC-2006-001, CMS-TDR-8-1, 2006. [Online]. Available: https://cds.cern.ch/record/922757

[12] CMS Collaboration, "CMSSW on Github," 2024, accessed: 2024-03-19. [Online]. Available: http://cms-sw.github.io/

[13] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *arXiv preprint arXiv:2103.13630*, 2021.

[14] Z. Yao, A. Gholami, K. Keutzer, and M. W. Mahoney, "Pyhessian: Neural networks through the lens of the hessian," in *2020 IEEE international conference on big data (Big data)*. IEEE, 2020, pp. 581–590.

[15] Y. Yang, L. Hodgkinson, R. Theisen, J. Zou, J. E. Gonzalez, K. Ramchandran, and M. W. Mahoney, "Taxonomizing local versus global structure in neural network loss landscapes," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 722–18 733, 2021.

[16] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International Conference on Machine Learning*. PMLR, 2019, p. 3519.

[17] N. J. Schaub and N. Hotaling, "Assessing intelligence in artificial neural networks," 2020.

[18] O. Weng, A. Meza, Q. Bock, B. Hawks, J. Campos, J. M. Duarte, N. Tran, and R. Kastner, "Fkeras: A sensitivity analysis tool for edge neural networks," in *Fast Machine Learning for Science Workshop at ICCAD 2023*, 2023.

[19] J. Hoffman, D. A. Roberts, and S. Yaida, "Robust learning with jacobian regularization," 2019.

[20] J. Diffenderfer, B. Bartoldson, S. Chaganti, J. Zhang, and B. Kailkhura, "A winning hand: Compressing deep networks can improve out-of-distribution robustness," *Advances in neural information processing systems*, vol. 34, pp. 664–676, 2021.